Documentation for the Spliced Leader ADDition (SLADD) program package for SLT analysis

Target audience

This is primarily written for someone with a modest basic knowledge of POSIX operating systems, i.e. who is able to create, inspect and organise files, links and directories, and has a workable understanding of the concepts of processes, shells, hierarcical file systems and environment variables. But it should also be useful as some kind of starting point for a more advanced user/developer, or just for getting a feel for how the pipeline works.

Prerequisites for full useability

- a somewhat POSIX compliant operating system (e.g. Linux, MacOSX)
- a working bash interpreter (available in almost all of the above)
- a working perl installation (perl 5.8.8 and 5.8.9 tested, but any perl5 should be fine, and is available in almost all of the above)
- a working installation of R (http://www.r-project.org/)
- The **BioPerl** library (http://www.bioperl.org, tested with bioperl-1.6.0).
- The BIO SAGE Comparison perl module (http://search.cpan.org/~SCOTTZED/Bio-SAGE-Comparison-1.00/, tested with 1.00)
- The text-NSP perl module, for an implementation of Fishers exact test. (http://search.cpan.org/~tpederse/Text-NSP-1.11, Tested with 1.09 and 1.11.)
- the short read alignment program **Maq** http://maq.sourceforge.net/
- seqlogo, from the Weblogo package (http://weblogo.berkeley.edu/)

At unibe, installations have been made on
Ubelix (Linux cluster),
izbhouston (MacOSX) and
izbalicante (Linux).
See the **nilsson** and **daniel** home directories for details.

Conventions

A couple of different fonts are used in this tutorial.

`uname -a; uptime; df -h; pwd`

Fixed space courier for shell commands to type at the terminal prompt.

/root/.[<shell>_]profile

Chalkboard for filenames. [] for optional parts, <> for variable content. Also used for file contents and environment variables.

Configuring the package upon installation at a new host computer

Obtain and extract the package.

`tar zxf sladd.releasedate.tgz`

Ensure all prerequisites are installed, note the paths for maq and seqlogo, for setting the appropriate environment variables, either for each run or for each user in their .profile. See the section on running the pipeline for details.

Ensure that the standard POSIX components such as grep, awk, head and cut are on the users search path, and bash is present or linked from /bin/bash, as well as perl from /usr/bin/perl. This is the typical setup for MacOSX and Linux. If you have a local BioPerl installation, rather than a system wide one, ensure that perl can find it, e.g. by pointing your PERL5LIB environment variable to it.

Preparing sequencing data

Previous batches of sequences have arrived as fastq format (http://en.wikipedia.org/wiki/FASTQ_format) read files, separated according to size. You will first need to decide on an insert size cutoff, and then concatenate the corresponding size files to one big readfile for each library.

```
cat Size_2[456789]*fastq Size_3[0123456789]* Size_4[0123456789]*
Size_5[0123456789]* Size_6[0123456789]* Size_7[0123456789]* > GDG-
1.Size24andup.fastq
```

The **naming convention** is libname.sizefilterdesc.fastq. This is now your tagfile.fastq.

Preparing template reference sequence data

Obtain template reference sequence fasta sequence and genome annotation gff files from an appropriate source, e.g. the Tr iTrypDB downloads area (http://tritrypdb.org/common/downloads).

```
wget http://tritrypdb.org/common/downloads/release-
2.0/Tbrucei/TbruceiTreu927Genomic_TriTrypDB-2.0.fasta

wget http://tritrypdb.org/common/downloads/release-
2.0/Tbrucei/TbruceiTreu927_TriTrypDB-2.0.gff
```

Rename or relink so that the pipeline can find files named according to the convention **refsequencename.fasta** and **refsequencename.gff** in the same directory

```
ln -s TbruceiTreu927Genomic_TriTrypDB-2.0.fasta
TbruceiTreu927_TriTrypDB-2.0.fasta
```

Running the main mapping and analysis pipeline

The pipeline is run in the directory where you store your datafiles. It generates a considerable number of files, including leftover temporary files, partial results files in addition to your desired results. It is advised to use separate directories for each compartive project. The pipeline is sensitive to timestamps. If a rawdata file is replaced, and the pipeline rerun, results files older than the new rawdata file (or significant intermediate results file, or generating script) will be replaced with new ones. That is a nifty feature, but nothing to worry about that right now. It may come in handy if you need to change only some part of the project later.

Environment: Several bash environment variables influence the pipeline behaviour.
Here are the most prominent ones, and their default values in parentheses.

BINDIR (~/install/bin)
: Set this to where your copy of the pipeline scripts is located.

MAQBIN (~/install/maq-0.7.1/maq)
: Set this to where your MAQ alignment binary is located.

SEQLOGOBIN (~/install/weblogo/seqlogo)

Set this to where your seqlogo script resides.

single (yes)
Do worry about producing a single mapper/high confidence set.

alqt (30)
Threshold for alignment quality for inclusion in the higher confidence set. Keep above 1 to keep multimappers out of the low confidence set.

runuorf (no)
Run uORF and N-terminal extension analyses (time consuming).

uorfutrcutofflen (2000)
Introduce a cutoff on the number of bases of UTR sequence to search for uORFs in.

utrcutoff (0)
Introduce a cutoff on the number of bases of UTR sequence in the gene attribution step. 0 means no restriction, i.e. arbitrarily long UTRs are allowed.

plotsplicesites (yes)
Do worry about extracting sequences surrounding splicesites and draw pretty logos for them.

The pipeline itself then takes **tagfile.fastq** and **refsequencename.fasta** for mandatory arguments. A third, optional parameter. Set this to whatever your target gene features are named in the **refsequencename.gff**.

```
run_maq.sh <tagfile.fastq> <refereneceseguencename.fasta> [gene_feature_type(gene)]
```

Assuming you use the bash shell interpreter, you would set these using `export`. On Ubelix, you will need to submit a batch job to the queue. Here is what a typical batch file would look like:

```
#!/bin/bash

export
PERL5LIB=/home/ubelix/izb/nilsson/lib/lib64/perl5/site_perl/5.8.8:/home/ubelix/izb/nil
sson/src/sladd

export BINDIR=~/src/sladd
export MAQBIN=~/bin/maq
export SEQLOGOBIN=~/src/weblogo/seqlogo
export single=yes
export uorfutrcutofflen=2000
export runuorf=yes
export plotsplicesites=yes

$BINDIR/run_maq.sh GDG-4.size24andup.fastq Tbrucei_TriTrypDB-1.1.fasta mRNA
```

filename: sladdlampa-gdg-4.sh

Please refer to the qsub manual/the ubelix user pages for details on how to submit the job.
On your own computer, or one of the smaller linux/osx boxes mentioned, you could just go with

```
export BINDIR=~daniel/src/sladd
```

```
export MAQBIN=~daniel/bin/maq

$BINDIR/run_maq.sh GDG-1.Size24andup.fastq
TbruceiTreu927_TriTrypDB-2.0.fasta mRNA
```

This will generate a set of useful (and a set of rather useless ;-) files:

<tagfile>_vs_<refsequencename>.<maq>map.**counts.summary**

A summary file, containing most of the overall results and statistics from the library. Viewable in any text editor.

<tagfile>_vs_<refsequencename>.<maq>map**[.q<cutoff>].counts.tab**

Tabular files containing tag counts, splicesites and so on on a per gene basis. All q30 files contain the same as their counterpart except only using the high quality, single mapping tags. Can be viewed in any texteditor or imported into a spreadsheet program, but take care to mark comma separated columns (UTRlen, Sladdpos, ...) as text fields, otherwise the spreadsheet program could interprete these as very large numbers, with commas separating thousands, depending how you have your number locale set.

<tagfile>_vs_<refsequencename>.<maq>map**[.<chromosome>][.q<cutoff>].crunch.gff**

GFF3 file (http://www.sequenceontology.org/gff3.shtml) for loading onto gbrowse or display in your favourite annotation viewer (Artemis, AGUI). These are plain text files, and will open in a text editor as well if needed.
The file has genes tagged with raw and normalised counts as well as 5'UTR-lengths. The hallmark is only one entry per mapped tag, however these are tagged with the sequence of each mapping tag, as well as the start position and gene name of the allocated gene.
For gbrowse loading, note that only the individual chromosome split files are provided with gbrowse required chromosome length headers.

<tagfile>_vs_<refsequencename>.<maq>map**.[category].UTRlen[.genes].pdf**

UTR length histograms. Default categories include `all_pos_UTRs_lt2k` and `genes_with_major_UTR_lt2k`. The .genes files count each length once, the default weight with transcript counts.

<tagfile>_vs_<refsequencename>.<maq>map[.category].once.**splicesites.out.pdf**

Splicesite logos. Category is one of **[major/minor/internal/external].** Only the version with each individual site counted once is created by default. seqlogo v2 uses up a huge amount of memory for the complete per transcript version, which also is not TPM scaled. Run the `extract_splicesites.pl` without the -1 option to create full sets. Or even better, add a call to the ExtractSites bash function in `run_maq.sh` after the existing calls. E.g. `ExtractSites "all" ""` would do the trick.

Other files that you may be interested in include:

<tagfile>_vs_<refequencename>.<maq>map**.gff**

GFF3 file containing raw mapping results – much data. Also available in high quality version (if enabled), and both of those also split into per chromosome.

<tagfile>_vs_<refequencename>.<maq>map**.gene.gff**

Each gene is provided with tags for the individual mapped reads allocated, and individual tags are tagged with the allocated gene. GFF3 and in principle gbrowse compliant, but will

need quite some database and rendering power.

<tagfile>_vs_<refequencename>.<maq>map.**antisense.log**
<tagfile>_vs_<refequencename>.<maq>map.**antisense.gff**
Antisense splice sites, per chromosome counts overview in the log and details in the gff file.

<tagfile>_vs_<refequencename>.<maq>map**.uorf.tab**
When uORF analysis is enabled, this file provides a count of uORFs and N-term-extension sites for each gene, in addition to the information provided in the ordinary **.tab** files. It has a column showing if the major UTR has uORFs, and one for the major SL add position, not only major UTR len.

<tagfile>_vs_<refequencename>.<maq>map[.q<cutoff>]**.counts.<subcategory>.tab**
Where subcategory is one of genes_with_major_site_internal, genes_with_only_internal_sites, genes_with_shortest_UTR_gt2k, genes_with_major_UTR_gt2k, and genes_with_major_UTR_lt2k, and these also havetheir high quality q[cutoff] counterparts. Hopefully selfexplanatory subcategories of genes in the standard counts.tab format. Useful for annotation updates and trouble shooting.

Preparing a comparison between libraries

If you will be running the comparison in the same directory as the mapping, you can procede directly to constructing a comparison file. If you do not, please first copy the counts.tab files you will want to use, along with the tagfile.fastq files to the new directory. Do consider soft or hard links to these, as they can be somewhat big. And this part of the pipeline only really uses them to calculate the library sizes.

The main component in a comparison is an orderded list of the counts tabs for the libraries you want to compare. Line/library order is important for image generation, tab file column order, the order in each pairwise comparison, etc. The first column has the counts tab file name, then follows a tab and the name of the library as you want it to appear on tables, figures etc.

GDG-9.Size23andup_vs_Tbrucei_TriTrypDB-1.1.maqmap.counts.tab	LongSlender
GDG-10.Size23andup_vs_Tbrucei_TriTrypDB-1.1.maqmap.counts.tab	ShortStumpy
GDG-7.Size23andup_vs_Tbrucei_TriTrypDB-1.1.maqmap.counts.tab	PC

filename: slc.comparison.list

Running a comparison between libraries

run_comparison.sh <comparison.list> [gene_feature_type(gene)] [kegg_organism(tbr)]

The comparison pipeline is run much like the one for mapping individual libraries. The optional gene feature name should be the same as for the mapping pipeline. The KEGG organism identifiers are as they appear on the KEGG website (http://www.genome.jp/kegg/pathway.html) and ftp-site.

The script takes an additional environment variable:

usekegg (no)
Run pathway analyses. You will need an internet connection, and some luck in order that the KEGG people have not once again changed their pathway file formats.

Then the script is run:

```
$BINDIR/run_comparison.sh slc.comparison.list tbr
```

Again, you will find a number of results files after the computation finishes.

<libname>_vs_<libname>.<alignmentprogram>.map.counts.compare

Pairwise comparison files. Contains four sections, heralded by lines starting with ***. **Differential splicing:** significantly different (P < 1e-5) major alternative splice sites. **Alternative splicing:** genes with major splice site with <60% of the total counts for that gene. **Up significantly:** significantly (P < 1e-5) upregulated genes (up in A.maqmap.count.tab) ordered by fold change. **Down significantly:** significantly (P < 1e-5) upregulated genes (up in B.maqmap.count.tab) ordered by fold change. The four tables have separate table headers, starting with **###** and some have brief summaries starting with **===.** The last two tables are also available on their own as one .**reg** file.

<comparison.list>**[.sign].norm.tab**

Tab separated list of tag counts for all genes (significantly changed only in the .sign files). Useful for inspecting details on ones favourite genes or chromosome using a text editor, import into a spreadsheet program or further analysis, clustering and heatmap plotting with e.g. MeV (http://www.tm4.org/) or bioconductor (http://www.bioconductor.org).

<comparison.list>**.norm.scatterall.pdf**

Scatterplot of all libraries against all. Genes significantly different between any pair of stages only shown on the subdiagonal.

<libtabfilename>**.altsplicestat.tab**

Interesting alternatively spliced genes, filtered according to criteria as shown in the lines begining with **===.** Available on a per library base.

pathway.summary.tab

Tag counts summed over each KEGG pathway for the organism.

pathway.**foldavg.**summary.tab

Relative tag counts over each KEGG pathway for the organism.

<comparison.list>.**summary**

Number of significantly regulated genes per pairwise comparison.

RNAseq mapping

This is not expected to be run too often again, but the interface is highly similar to the SLT mapping.

run_rnaseqmap.sh <tagfile.fastq> <referenecesequencename.fasta> [gene_feature_type(gene)]

It accepts fewer environment variable options, as those pertaining to uORF mapping and splicesite drawing are not applicable.

Only the core set of gff files and tables are produced. A novelty is a set of .wig files, giving per column tag densities for loading onto e.g. gbrowse.

Questions

Questions can be forwarded to daniel.nilsson@izb.unibe.ch or daniel.k.nilsson@gmail.com.